

Angular 1

17/01/2020

Paolo Cargnin

Cos'è Angular



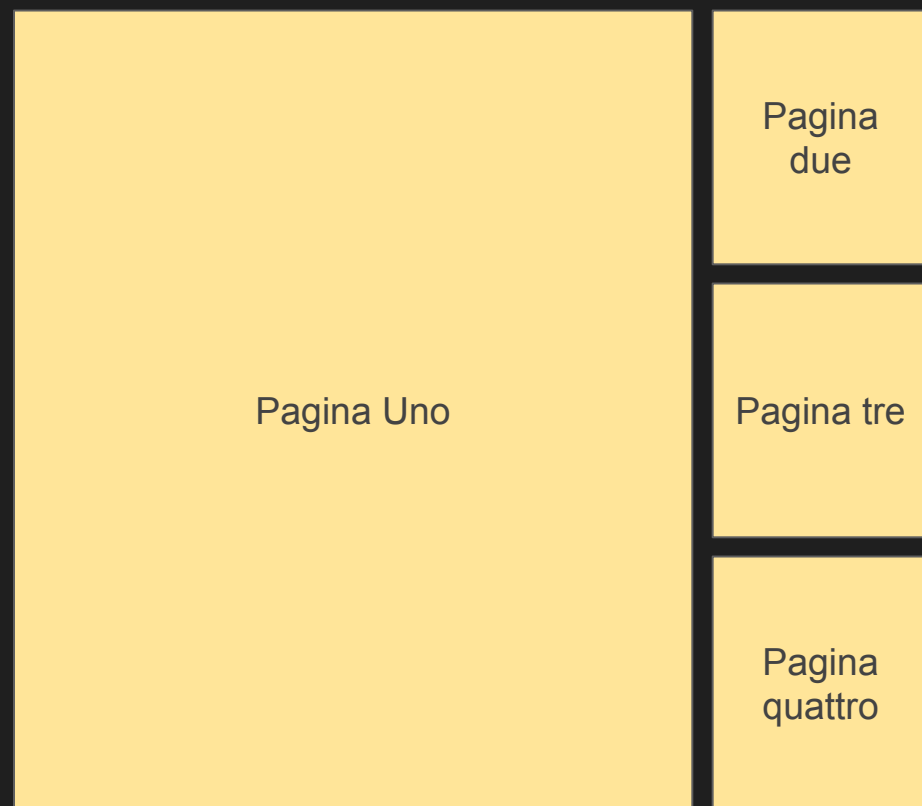
Un framework JavaScript che ci permette di creare Single Page applications

Cos'è una Single Page Application (SPA)?

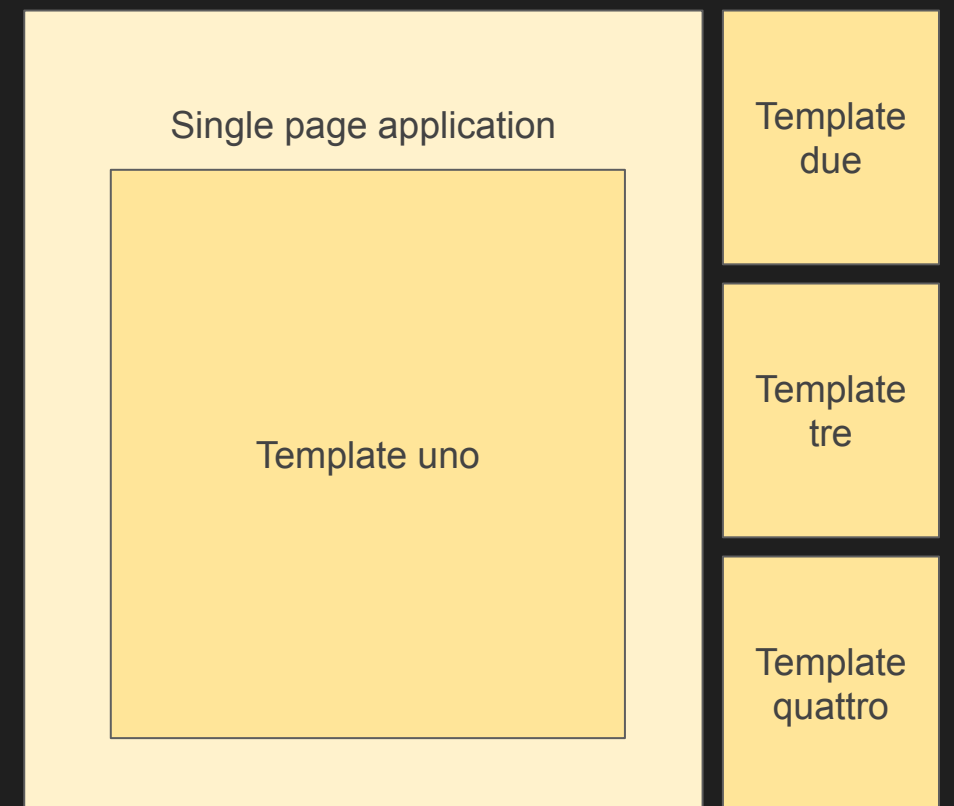
Un applicativo web, che funziona senza il bisogno di aggiornare la pagina.

Dinamicamente viene aggiornata la pagina (e l'url) in cui è presente l'utente.

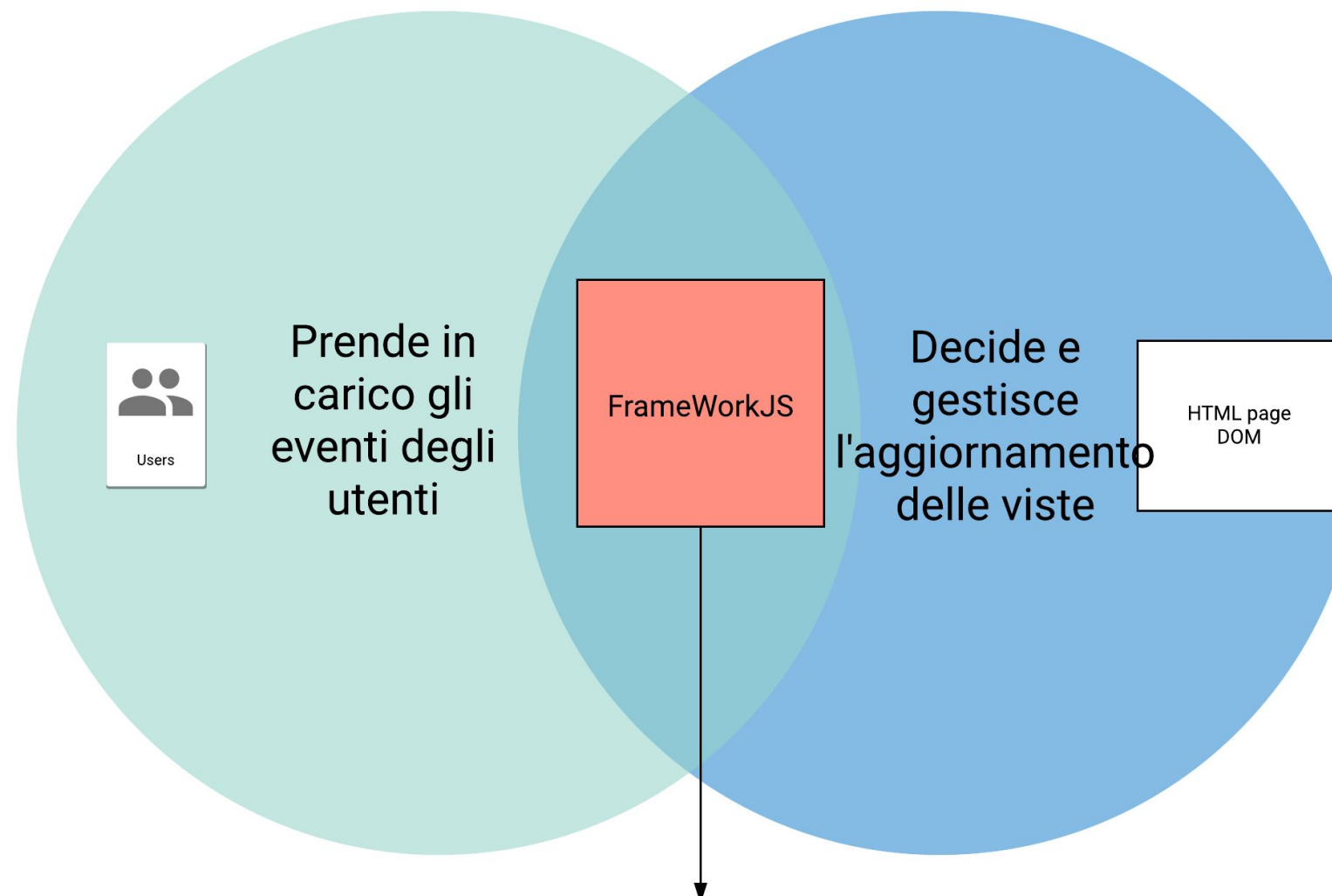
Applicativi tradizionali



SPAs



Funzionamento di un FrameWork JS per SPAs



- Gestisce il codice e la logica dell'applicativo
- Convalida i dati dell'utente
- Ottimizza l'aggiornamento dell'HTML
- Gestisce gli URL della pagina ecc ecc ecc

Angular, un po' di storia

Angular (ver 1) o *Angular Js*

Il primo framework JS **Completo**

Angular 2

Completamente riscritto.

Angular 3 non esiste. Ogni sei mesi esce
una nuova versione
(con poche modifiche dalla precedente).

oggi impariamo **Angular 8**

Iniziamo ad usarlo:

angular-cli

- Gestione automatica di tutto il progetto
- Gestione delle dipendenze:
 - *packages.json*
 - *Typescript*
 - *webapck*
 - *ecc ecc*

Installiamolo globalmente dal terminale

```
npm i -g @angular/cli@latest
```

Poi ci basterà eseguire:

```
ng new my-first-app
```

Ci chiederà alcune domande

Routing

Per gestire le rotte (gli URL)

How you are going to write CSS

Per scrivere css

Vedremo tutto a tempo debito.

Ok Basta slide. apriamo l'editor

`code my-first-app`

Se non vi apre Visual Studio Code,

fatele voi

Prima sessione:

1. Eseguiamo ng serve da VSC
2. Guardiamo il codice: cartella src
3. Guardiamo il codice: cartella App
4. Eliminiamo il codice e mettiamo un input element
5. `import { FormsModule } from '@angular/forms';`
6. Guardiamo il codice: il file index.html

Struttura del corso

1° lezione

Introduzione
Componenti e data binding
base
+ storia dei JS frameworks

2° lezione

Componenti e data binding
in applicazioni avanzate

3° lezione

Direttive
+ Setup progetto del corso

4° lezione

Services e dependency
injection

5° lezione

Routing
Observables

6° lezione

Forms
Pipes

7° lezione

HTTP
Autenticazione

8° lezione

Ottimizzazioni e ngModules
Interface Development
UI frameworks

Iniziamo gli esercizi

1 – Inseriamo **Bootstrap**

Framework CSS (scss in realtà) che oggi ci
velocizza il lavoro
(e ci insegna un po' di CSS)

Ci sono framework specifici per Angular,
ma li vedremo più avanti

da VSC:

```
npm i -s bootstrap
```

- Modifichiamo il file angular.json
 - svuotiamo il progetto

Componenti

Blocchi di codice che compongono tra di loro l'interfaccia utente. Un'applicazione Angular è un albero di componenti

- Possono essere riutilizzabili
- Sono inizializzati tramite il decorator `@Component`

Componenti

HOME

ABOUT

Primo esercizio:

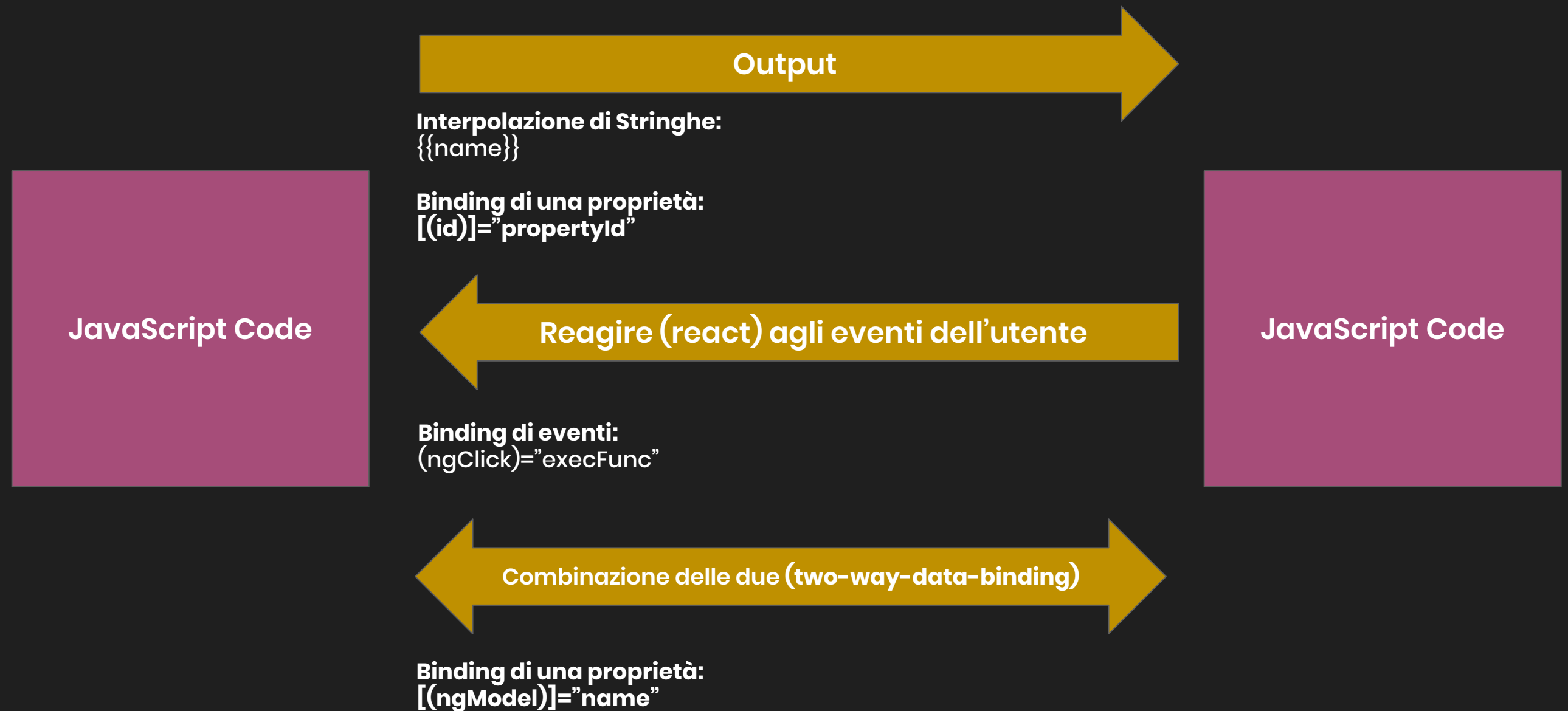
001-basics-002-components

Data binding

Come il nostro javascript
(la nostra app)

Comunica con il template html
(quello che vede l'utente)

Flusso di dati



Secondo esercizio:

001-basics-003-data-binding

Directives

Istruzioni nel DOM per interagire con i singoli componenti

Già presenti in angular
(built in)

ngIf
ngClass
ngStyle
ngFor
...

Creabili da noi
(custom)

Terzo esercizio:

001-basics-004-directives

Storia dei moderni framework

1. JQuery – Semplificare JavaScript
2. Framework che seguono un PATTERN – AngularJs
3. Virtual DOM e componenti
4. Webpack a chiudere il tutto

JQuery - Semplificare JavaScript

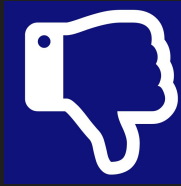


- Selezione
- Manipolazione del DOM
- Gestione degli Eventi

```
//close modal & deatroy carousel
$('.close-gallery').click(function() {
  $('.modal-gallery').addClass('modal-gallery-close');
  setTimeout(function(){
    $('.recive-gallery').slick('destroy');
  }, 500);
});
```

A- storia dei moderni framework

JQuery - Semplificare JavaScript



- Fetch di chiamate AJAX
- Gestione degli INPUT (validazione, getter e setter)
- Gestione di eventi poco performante (scroll, resize, view update)
- Gestione delle variabili Globali

JQuery - Semplificare JavaScript



```
$('.nl-form').each(function(){
    var privacyCheck = $(this).find('#privacy_check');
    var txtNormal = privacyCheck.parent().find('.text-white.small').html();
    var messagePrivacy = $('html').attr('lang') === 'it-IT' ? 'Checkbox obbligatoria' : 'Mandatory checkbox';
    var cityreuire = $(this).find('input#city').length > 0;
    var professionreuire = $(this).find('input#profession').length > 0;
    $(this).validate({
        rules: {
        },
        messages: {
        },
        // grecaptcha.ready(function() {
        submitHandler: function(form,e) {
            e.preventDefault();
            if (! privacyCheck.prop('checked') == true){
                privacyCheck.parent().find('.text-white.small').html(`${txtNormal} - ${messagePrivacy}`);
                return false;
            }else{
                privacyCheck.parent().find('.text-white.small').html(txtNormal);
            }
            $(form).parent().addClass('loading');
        },
    });
});
```

Framework che seguono un PATTERN - AngularJs



- Two way data binding tra CONTROLLER e DOM
- Gestione di variabili globali e locali
- Template Engine integrato
- Gestione degli url (Routing)

```
<!doctype html>
<html ng-app>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.7.8/angular.min.js"></script>
  </head>
  <body>
    <div ng-controller="myApp">
      <label>Name:</label>
      <input type="text" ng-model="yourName" placeholder="Enter a name here">
      <hr>
      <h1>Hello {{yourName}}!</h1>
    </div>
  </body>
</html>
```

```
angular.module('todoApp', [])
  .controller('myApp', function() {
    this.yourName = 'Paolo Cargnin'
  });
```

Framework che seguono un PATTERN - AngularJs



- Gestione di eventi poco performante (scroll, resize, view update)
- Manipolazione del DOM
- Applicativo globale, da assegnare a tutta la soluzione

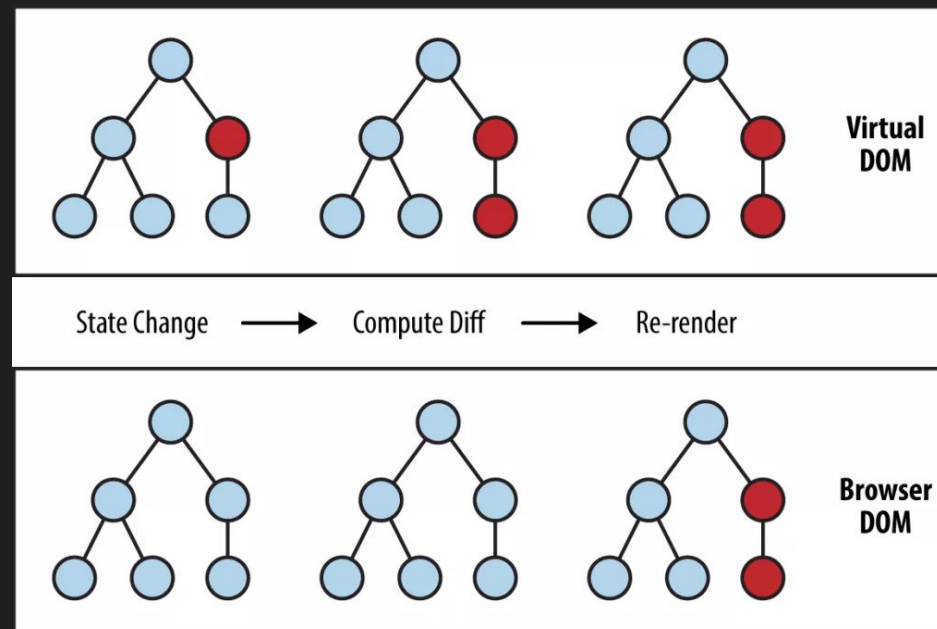
Inoltre: Sarebbe impossibile pensare insegnare angularJs nelle scuole.

```
angular.module('todoApp', [])  
  .controller('myApp', function() {  
    this.yourName = 'Paolo Cargnin'  
    $('input[ng-model="yourName"]').val('Giamba Gennari'  
  
    console.log(this.yourName) // -> Paolo Cargnin  
  
    $timeout(function () {  
      console.log(this.yourName) // -> Giamba Gennari  
    });  
  });
```

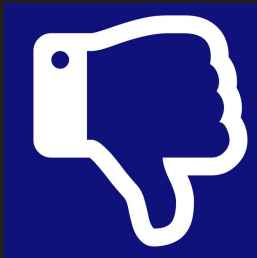
Virtual DOM e componenti - REACT



- ONE way data binding
- Manipolazione del DOM efficace
- Tutti i vantaggi di un framework per applicativi complessi
- Può essere usato anche in porzioni della pagina web



Virtual DOM e componenti - REACT



- Curva d'apprendimento
- Una dipendenza specifica per ogni compito (axios, react-router, Immutable ecc ecc)

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.scss';
import App from './components/App/';

import registerServiceWorker from './registerServiceWorker';

import { createStore, applyMiddleware } from 'redux';
import thunk from 'redux-thunk';
import { loadTranslations, setLocale, syncTranslationWithStore } from 'react-redux-i18n';
import reducers from './reducers';

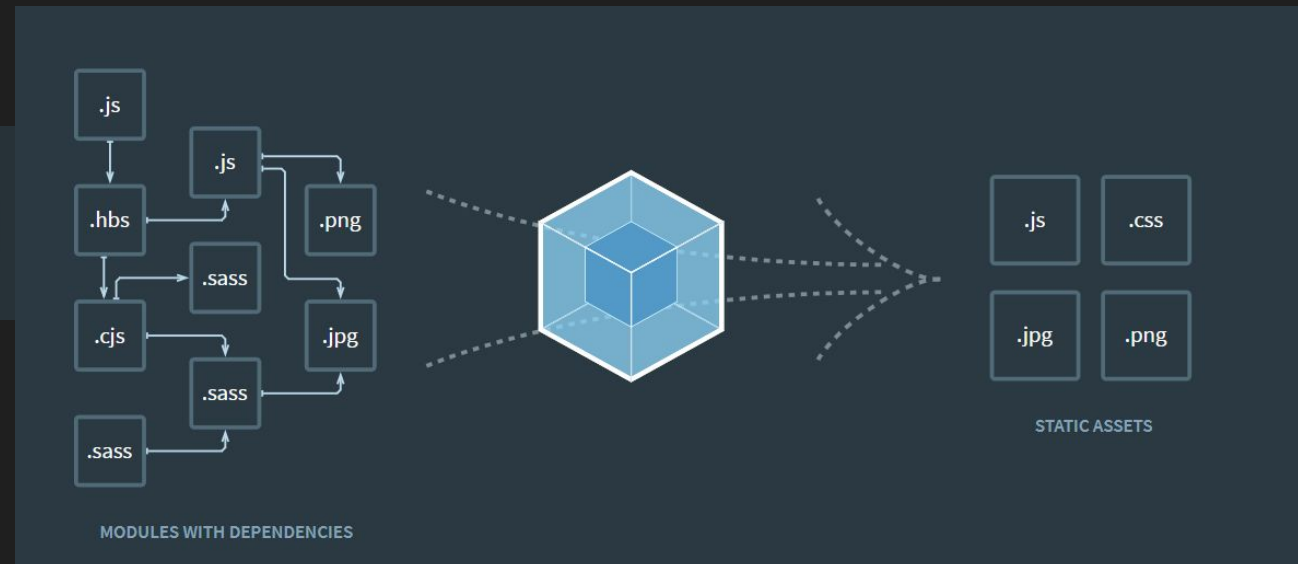
import translationsObject from './languages';
import { Provider } from 'react-redux';
```


Webpack, export (default e named) e import

```
import SlideViewer from './SlideViewer.vue'

import '@assets/presentations/js/plyr.min.js'
import '@assets/presentations/scss/plyr/plyr.scss'
import '@assets/presentations/scss/main.scss'

export default {
  export const postHeaderConfig = {
    headers: {'Content-Type': 'text/xml'}
  };
}
```



Rimangono ancora un po' di cose migliorabili

- La gestione delle dipendenze è un po' confusionale
 - Inizializzare un progetto non è esattamente semplice
- Validare e conoscere tutti i framework e relativo utilizzo è impossibile

Diversi tipi di soluzioni con diversi risvolti

- TypeScript
- Angular Cli, Vue Cli ecc
 - Babel
 - JSX
 - ecc ecc